11-7-2008

Development of a Dual-Handed Haptic Assembly System: SHARP

Abhishek Seth
Iowa State University

Hai-Jun Su
University of Maryland, Baltimore County

Judy M. Vance
Iowa State University, jmvance@iastate.edu

Development of a Dual-Handed Haptic Assembly System: SHARP

Abstract

Virtual reality (VR) technology holds promise as a virtual prototyping (VP) tool for mechanical assembly; however, several developmental challenges still need to be addressed before VP applications can successfully be integrated into the product realization process. This paper describes the development of System for Haptic Assembly and Realistic Prototyping (SHARP), a portable virtual assembly system. SHARP uses physics-based modeling for simulating realistic part-to-part and hand-to-part interactions in virtual environments. A dual-handed haptic interface for a realistic part interaction using the PHANToM® haptic devices is presented. The capability of creating subassemblies enhances the application's ability to handle a wide variety of assembly scenarios at the part level as well as at the subassembly level. Swept volumes are implemented for addressing maintainability issues, and a network module is added for communicating with different VR systems at dispersed geographic locations. Support for various types of VR systems allows an easy integration of SHARP into the product realization process, resulting in faster product development, faster identification of assembly and design issues, and a more efficient and less costly product design process.

Disciplines

Mechanical Engineering

Comments

This article is from *Journal of Computing and Information Science in Engineering* 8 (2008): 044502, doi:10.1115/1.3006306. Posted with permission.

Development of a Dual-Handed Haptic Assembly System: SHARP

Abhishek Seth
Ph.D.
Department of Mechanical Engineering,
Virtual Reality Applications Center,
Iowa State University,
Ames, IA 50011
e-mail: abhiseth@vrac.iastate.edu

Hai-Jun Su
Ph.D.
Department of Mechanical Engineering,
University of Maryland,
Baltimore County,
Baltimore, MD 21250
e-mail: haijun@umbc.edu

Judy M. Vance
Ph.D.
Fellow ASME
Department of Mechanical Engineering,
Virtual Reality Applications Center,
Iowa State University,
Ames, IA 50011
e-mail: jmvance@iastate.edu

Virtual reality (VR) technology holds promise as a virtual prototyping (VP) tool for mechanical assembly; however, several developmental challenges still need to be addressed before VP applications can successfully be integrated into the product realization process. This paper describes the development of System for Haptic Assembly and Realistic Prototyping (SHARP), a portable virtual assembly system. SHARP uses physics-based modeling for simulating realistic part-to-part and hand-to-part interactions in virtual environments. A dual-handed haptic interface for a realistic part interaction using the PHANToM® haptic devices is presented. The capability of creating subassemblies enhances the application's ability to handle a wide variety of assembly scenarios at the part level as well as at the subassembly level. Swept volumes are implemented for addressing maintainability issues, and a network module is added for communicating with different VR systems at dispersed geographic locations. Support for various types of VR systems allows an easy integration of SHARP into the product realization process, resulting in faster product development, faster identification of assembly and design issues, and a more efficient and less costly product design process.
[DOI: 10.1115/1.3006306]

Keywords: haptics, virtual reality, virtual prototyping, human-computer interaction, virtual assembly, swept volumes, physics-based modeling

Contributed by the Engineering Simulation and Visualization Committee for publication in the Journal of Computing and Information Science in Engineering. Manuscript received November 2, 2007; final manuscript received September 1, 2008; published online November 7, 2008. Guest Editors: J. Oliver, M. Omalley, and K. Kesavadas.

1 Introduction

VR technology is gaining popularity as an engineering design tool and is increasingly used as a digital test-bed for early prototypes. VR simulations are used as a tool during the product design process to evaluate design alternatives for assembly, manufacturability, maintainability, etc. However, in order to use digital product models for advanced evaluations, a virtual prototype must exhibit a behavior that is very similar to physical models. For instance, the digital environment should provide the same level of human/product interaction, allow for similar testing scenarios, and accurately reflect the evaluations obtained when using physical models. Sensory evaluations such as visual, haptic (force feedback), and auditory feedback are also important to accurately evaluate product performance. VR techniques are used throughout the design process to simulate different stages of product realization, i.e., evaluating multiple design concepts, manufacturing processes, assembly process planning, plant layout, maintenance evaluations, etc.

A virtual assembly (VA) system as proposed in this paper will empower future engineers with a platform that will allow them to visualize and realistically interact with multiple design alternatives during conceptual stages before physical prototypes are built. Such a system will facilitate identification of product/process design errors during early stages of product development where major changes are still feasible. Thus, it will reduce unforeseen problems that arise during later stages of the product life cycle, consequently saving both time and money while improving product quality.

2 Research Challenges

During the past two decades, VR technology has evolved to a level where immersive virtual walkthroughs and data visualization simulations have become commonplace. Prototyping assembly/disassembly processes in virtual environments present a much more challenging problem because they require frequent, direct, and intuitive human interactions with virtual product models. To simulate simple real world assembly tasks in a virtual environment, a VA system must include the following features (Table 1): graphical visualization, which provides visual feedback; object behavior modeling, which simulates the physical interaction (dynamics, collision, and friction) between part-part and hand-part; haptic force feedback, which allows the worker to feel contacts that occur between parts; and dual-handed assembly. In addition, capabilities such as subassembly creation, part joining methods, and interaction with tools and fixtures also form core components of the simulation. Prominent challenges in this field are classified into four categories and are elaborated below.

2.1 Graphic Visualization. Immersive and realistic graphical visualization is important for tasks such as part picking and placement, which require understanding 3D spatial relationships among computer-aided design (CAD) models. Stereo visualization and high level-of-detail (LOD) product models are critical in providing an accurate representation of the real world assembly scenarios. CAD assemblies containing thousands of parts present problems for interactive visualization due to the "excessive number of polygons and number of objects that are created" [1].

2.2 Collision Detection. Another critical challenge in creating VA simulations is accurately modeling the physical behavior of parts. Collision detection algorithms are frequently used to prevent part interpenetration during assembly. Mechanical assembly scenarios demand an accurate collision detection among arbitrarily complex (nonconvex) CAD geometries. In VA simulations where real-time update rates are critical, performing a fast and accurate collision detection among dynamic objects is a challenging problem.

Table 1 VA research challenges

Features	Challenges
Graphical visualization	• High level-of-detail (LOD) product models • Low cost immersive VR systems • Support for multiple VR systems
Realistic object behavior among complex CAD models	• Physics (dynamics, friction, etc.) modeling of CAD models with complex topology • Real-time collision detection with high precision • Dynamic interaction between part-part and hand-part • Minimize data translation between CAD and VR
Haptic force feedback	• Haptic rendering rate • Feedback part-part collision force natural to the operator
Dual-handed assembly	• Simulate natural part manipulation • Maintain physics and haptic update rates
Subassemblies/ disassemblies	• Update data structure, affecting part interaction and haptic force calculation
Assembly planning	• Generate data (swept volume, assembly sequence, etc.) useful for engineering practice

2.3 Physics-Based Modeling. Once collisions are detected in the environment, physics-based modeling algorithms are needed to compute the subsequent part trajectories. Such algorithms [2–5] solve equations of motion of objects at each time step based on forces and torques that act upon the objects. All these have different limitations, such as modeling accuracy, handling stable and simultaneous contacts, large computation time when many contacts occur, and system instabilities leading to stiff equations, which are numerically intractable [6]. Approximate model representations are generally used to maintain interactive update rates. Due to such problems, very few VA applications rely solely on physical constraint simulation to perform assembly [7–9].

2.4 Haptic Interaction. In manipulation intensive tasks such as assembly, haptic force can help a designer feel and better understand the geometry of virtual objects. Haptic devices require a high update rate ($\sim$1000 Hz) to guarantee force continuity. Hence, the real challenge is to perform collision and physics computations upon large, arbitrary, and complex CAD data sets at haptic update rates. Further, handling multiple haptic devices simultaneously makes the problem even more complicated.

3 Background

Several research groups have attempted to address the challenges of VA using existing technologies. Stereo viewing, head tracking, and instrumented glove interaction are all common components of many VA applications [10–12]. Efforts have also been directed at interacting with complex CAD models [12–14]. Recently, haptic interaction has been integrated into many of these applications [13,15,16]. Haptic interaction provides force feedback to the user as an additional sensory input to aid in evaluating assembly tasks in the virtual environment.

The Inventor Virtual Assembly (IVY) system developed by Kuehne and Oliver [11] used IRIS OPEN INVENTOR graphics library that allowed designers to interactively verify and evaluate the assembly characteristics of components directly from a CAD package. Parts were selected using assembly hierarchy as collision detection was not supported by the system. A desktop-based system called Virtual Environment for Design for Assembly (VEDA) [17] used dual PHANToM® haptic devices to grasp CAD representations using the user's finder-tips. The system could only simulate interactions between 2D CAD representations. Coutee et al. [15] developed a similar desktop system called Haptic Integrated Dis/re-assembly Analysis (HIDRA). OPENGL was used for visualization on a 2D monitor, and V-CLIP in conjunction with Q-HULL and SWIFT++ were used for collision detection. The system had problems handling nonconvex CAD geometry and did not allow intuitive part manipulation.

Fröhlich et al. [7] developed an interactive VA system using physics-based modeling. The system used a Responsive Workbench for simulating bench assembly scenarios. Haptic feedback was now available, and the system encountered problems when several hundred collisions occurred simultaneously. Virtual Assembly Design Environment (VADE) developed by Jayaram et al. [12] used assembly constraints and transformation matrices imported from PRO/E to complete the assembly in VR. Two-handed assembly was simulated using CyberGlove devices. A physics-based algorithm with limited capabilities was added to VADE for simulating realistic part behavior [18]. However the system did not provide any haptic feedback. Bullinger et al. [19] developed an assembly planning system, which used an anthropometric computer modeling software package, to perform ergonomic evaluations during assembly. Fernando et al. [20] created a VA that used constraint-based modeling for assembly. The system used a constraint manager [14], which identified, applied, and deleted geometric constraints during assembly. Kim and Vance [10] utilized physics-based modeling to simulate realistic part behavior. The Network Haptic Environment (NHE) [13] was developed to facilitate collaborative assembly through the internet. The variety of computation capability of each node often caused inconsistency problems, which produced unrealistic haptic forces. Wan et al. [16] developed a multimodal CAVE™-based VA, which used geometric constraints for simulating part behavior. The users could feel the shape of digital models using the CyberGrasp haptic device. However no force feedback was available when parts collided. Brough et al. [21] developed a virtual assembly simulation for training related tasks. The focus of this work was on the cognitive aspects of training instead of realistic physics-based simulations. Garbaya and Zaldivar-Colado [22] created a physics-based VA system, which used a spring-damper model to provide the user with collision and grasping forces during the mating phase of an assembly operation. An experimental study concluded that user performance increased when interpart collision forces were rendered as compared with when only grasping forces were provided to the user.

4 Motivation

The focus of the work presented in this paper is to create a system that can address the challenges outlined and provide a successful solution to the VA problem. Once successful, the VA capability will provide the foundation for many useful virtual environments, including virtual process planning, task timing, workstation layout, tooling design, and integration of the immersive virtual environment with interactive discrete event programing. In addition, the results of this research will support further development of immersive offline training, maintenance, and serviceability prototyping.

Our intent is to develop and evaluate a system that spans various levels of VR hardware from desktop to full immersion in order to explore how all of these different VR interfaces might be used together to improve the design process. In this paper we present System for Haptic Assembly and Realistic Prototyping (SHARP). The following section describes the system configuration and methodology used for assembly/disassembly simulation in SHARP. Next, this paper will describe additional components, which expand SHARP's capabilities to expand the system's ability to address problems related to maintainability, training, and collaborative analysis using virtual environments. SHARP takes advantage of previous knowledge [12,13,15,16,23] and expands the

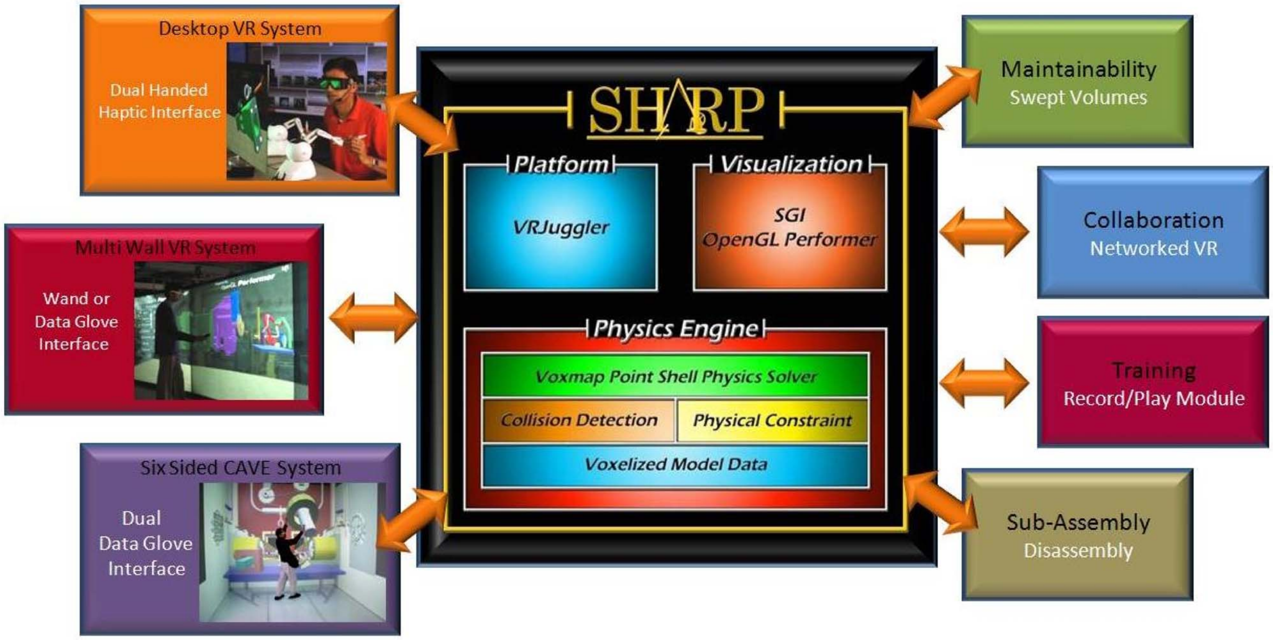

Fig. 1 SHARP system components and modules

functionality of VA to include dual-handed haptics, swept volume representation, subassembly modeling, and realistic part behavior.

5 SHARP: A System for Haptic Assembly and Realistic Prototyping

A VA system requires combining knowledge from multiple research areas such as VR, human-computer interaction, and engineering design. The three main components of a VA simulation consist of visual, behavioral, and interaction realism. Figure 1 describes the main components of the SHARP system. The system core consists of the platform, visualization, and physics behavior engine. The VRJUGGLER [24] open source library is used as an application platform for this research. VRJUGGLER hides many low level programing details required to develop, test, and run applications on different VR systems. This enables SHARP to be ported to different VR system configurations from desktop and power walls to immersive CAVE systems. To provide a realistic interaction with product models on desktop VR systems, a dual-handed haptic interface is developed. In fully immersive VR systems such as CAVE, multiple trackers are used to track the user's hands, and wireless 5DT data glove [25] devices are used for a dual-handed interaction. Gesture recognition is used for intuitive part grabbing. However these devices do not provide haptic feedback to the user. Various modules are developed to utilize the SHARP's core capabilities for maintainability, collaboration, and training purposes. These modules will be described in Sec. 8.

Realistic and detailed graphic representations are created in SHARP using optimized scene-graph-based data structures [26], which allow visualization of high LOD models along with their material properties and surface textures. The core of the VA system is the behavior engine that guides part movements as well as placement for assembly. SHARP computes physical constraints among contacting part surfaces in real time to accurately simulate real world assembly scenarios. The Voxmap Pointshell (VPS) software [3] is used for collision detection and physics-based modeling. VPS is chosen as the physics-based behavior engine for SHARP because

(1) VPS can operate on CAD models of complex geometry,
(2) VPS works well when there are a small number of moving objects in the virtual environment, and
(3) VPS is optimized for maintaining the haptic force update rate as high as 1000 Hz [27].

5.1 Model Preprocessing and Representation. Seamless integration of VA applications into the design process requires a frequent and efficient data exchange between CAD and VA systems. It is important to note that the system design proposed in this research supports direct data transfer from any CAD system with minimal preprocessing and does not rely on proprietary CAD toolkits and metadata for creating assembly scenarios. For every model in the scene, the system uses a graphic model representation and a physics model representation. A virtual object class is created, which holds both physics and graphics representations of each object.

Graphics. For graphic model representation (Fig. 2), .wrl, .iv, .3ds, .pfb, and several other generic CAD formats can be used. Every model node is assigned a transformation matrix that guides its position and orientation in the graphics world.

Physics. For physics computations, a standard .stl file format is used. The .stl file is parsed, and the triangle and normal information are loaded into a data structure. During the voxelization step, the set of triangular polygons read from the file is converted to the

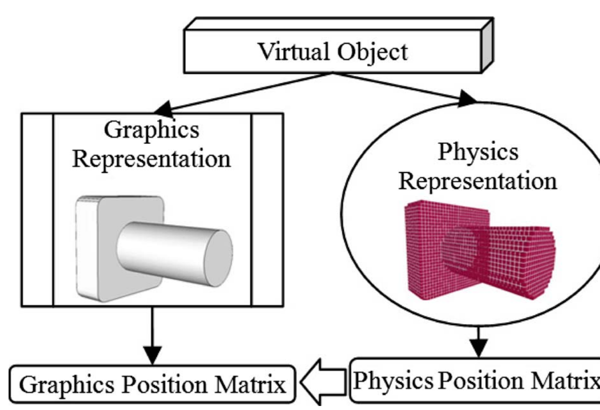

Fig. 2 Model data structure in SHARP

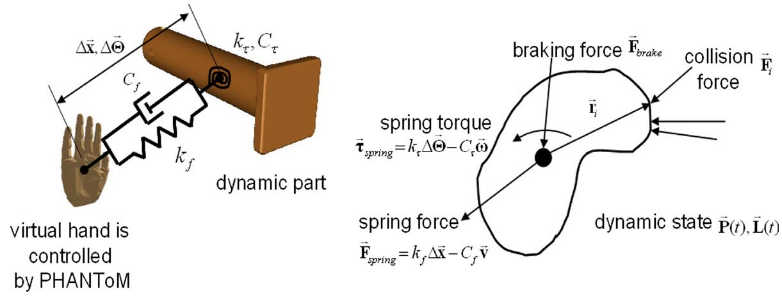

Fig. 3 Physics-based modeling in VPS

VPS spatial representation called voxmap. Physical properties such as mass, center of mass, and moment of inertia for each CAD model are then calculated by the system, completing the system initialization process.

5.1.1 Realistic Object Behavior. When developing a virtual environment that supports interactive manipulation and assembly of complex CAD objects, the greatest challenge is achieving realistic part behavior to manage the trade-off between object complexity and computational burden. Most often, an approximate geometric model is used for collision detection and force calculations [28]. A coarsely defined approximate model allows for fast, but inaccurate, collision and force calculations. Similarly, a model that closely approximates real geometry may contain unnecessary details which could prevent the system from maintaining interactive rates.

Each CAD model is discretized into a set of voxels (cubic elements) creating a "voxmap," which is used for collision detection and physics computation. A pointshell is created for the moving object, which consists of points located at the centers of each voxel element. When two objects collide with each other, VPS returns the contact force proportional to the penetration of the pointshell of the moving object into the voxmap of the static object.

The collision force $\mathbf{F}_i$ is proportional to the amount of penetration that one object has into the other object in the environment. The manipulated object is dynamic in nature, and its motion is subject to physics laws, more specifically rigid body dynamics. That is, given the dynamic state of a rigid body at time t, its motion must satisfy Eqs. (1) and (2),

$$\frac{d\mathbf{P}(t)}{dt} = \mathbf{F}_{\text{total}}(t) \tag{1}$$

$$\frac{d\mathbf{L}(t)}{dt} = \mathbf{M}_{\text{total}}(t) \tag{2}$$

where $\mathbf{P}(t)$ and $\mathbf{L}(t)$ are the linear and angular momenta of the rigid body and $\mathbf{F}_{\text{total}}(t)=\mathbf{F}_{\text{spring}}(t)+\Sigma\mathbf{F}_i+\mathbf{F}_{\text{brake}}$ and $\mathbf{M}_{\text{total}}(t)=\boldsymbol{\tau}_{\text{spring}}(t)+\Sigma\mathbf{r}_i\times\mathbf{F}_i$ are the total external force and moment exerted on the body, respectively. For our case, they are given by the sum of the force/torque applied by the virtual spring, the collision force applied by other objects, the damping force, and the braking force. The rigid body dynamics equation is solved using the VPS function "VpsPbmEvolve." After a collision occurs, the physics loop calculates subsequent model positions, which are used to update the graphics scene-graph. See Ref. [3] for more details regarding VPS methods. A careful selection of the amount of discretization of the VPS haptic model is needed in order to produce a representation that is sufficiently modeled so that tight tolerance parts can be assembled. SHARP allows for individual models to have different voxel sizes for managing the trade-off between accuracy and computation speed.

6 Dual-Handed Haptic Interface

Most VR applications require users to perform simple navigational tasks or launch preprogramed set of events during the simulation. Wands, joysticks, and other advanced wireless controllers have been successful in providing us with an effective interface for such applications. Manual assembly simulations, on the other hand, require users to use both their hands naturally to successfully simulate real world tasks.

6.1 Virtual Coupling. A well known "virtual coupling" method [29] is implemented in this research as a link between the haptic device and the virtual environment. Since this research uses impedance type haptic devices (which measure motion and display force), a virtual coupling is necessary to guarantee haptic rendering stability. When a user grasps a part, a virtual spring and damper system is attached between the part and the virtual hand (Fig. 3). The distance between the virtual hand and the manipulated object determines the spring force $\mathbf{F}_{\text{spring}}$ and torque $\boldsymbol{\tau}_{\text{spring}}(t)$ exerted on the object. Note that the spring force and torque also include the viscous force of the damping system. This spring force is sent to the haptic device for rendering. A nice feature of virtual coupling is that it allows the user direct and intuitive (rotational and translational) control over the manipulated object. In addition, it allows the capability to tweak the spring and damper constants independent of the physical simulation. Higher spring stiffness corresponds to sharper force feedback during collision; however it results in drag during free manipulation of objects.

6.2 Implementation. A single-handed haptic interface was initially created for SHARP, which provided users with force feedback whenever collisions occurred during the simulation [23]. All physics computations were performed in a separate high priority thread to get an optimal physics update rate (~1000 Hz) for haptic rendering.

A dual-handed simulation required expanding this system to support multiple hands in the environment. A new hand model data structure has been created in SHARP, which defines properties (haptic data, graphic data, hand position, control source, etc.) and states (colliding, grabbing, etc.) of each hand instance present in the scene. This provided the user the capability for simultaneous part manipulation using multiple hand instances. The system has to compute physical responses for each hand instance present in the scene during every physics frame. Thus, the physics update rate is halved every time a new hand instance is added. The graph in Fig. 4 shows the physics idle update rates for single- (~1000 Hz) and dual-handed (~500 Hz) configurations. It is important to note that the physics update rate is dependent on the CPU speed. However the haptics loop always runs at 1000 Hz.

For a very small change in part position between consecutive physics frames, the change in transmitted force will be unnoticeable to the user. The system takes advantage of this fact by continuing to render the last calculated force until new forces are

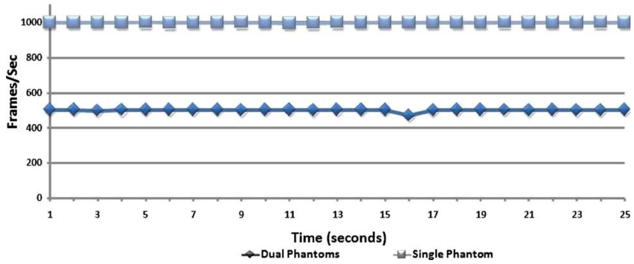

Fig. 4 Physics update rate for single and dual-handed configurations

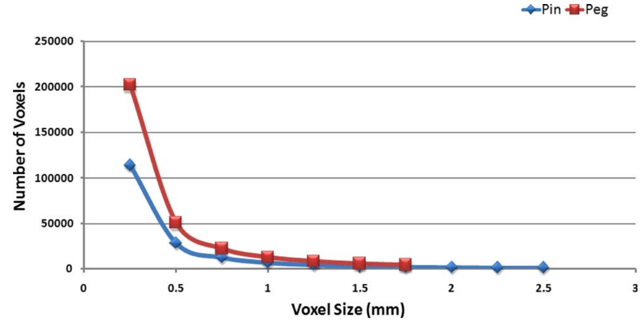

Fig. 7 Number of voxels versus voxel size

computed. We have found that this approach provides smooth forces with physics update rates as low as ~200 Hz (Fig. 5).

6.3 Mapping the Haptic Workspace. A series of transformations is used to map the haptic workspace of each haptic device in the virtual environment (Fig. 6). The original device coordinates (x_d, y_d, z_d) are transformed to account for the difference between the millimeter units of the device workspace and the default feet units used by VRJUGGLER (x_j, y_j, z_j). In addition, a transform is applied to appropriately scale the real haptic workspace (RHW) such that the virtual haptic workspace (VHW) is enlarged to represent the reach ability of an average human hand (x_h, y_h, z_h). As SHARP supports different PHANToM® haptic devices, this transform varies based on the RHW of each device. These coordinates are then multiplied by the camera matrix to generate VHW within camera view coordinates (x_c, y_c, z_c). This ensures that the VHW always stays within the user's view and also allows the user to move the VHW as he/she navigates the virtual environment.

After the initial development from the single-handed to the dual-handed configuration, both haptic devices were initialized such that they had the same VHW. During demonstrations at various conferences and public exhibits, users expressed difficulty in keeping track of the left and right hands within the environment due to completely overlapping VHWs.

To address this usability issue, the workspaces are shifted so that there is only a 30% overlap. This change helps users distinguish between their left and right hands in the application and allows a more realistic dual-handed interaction. Interacting with two hands and receiving force feedback, an operator can more realistically perform assembly tasks with the same dexterity as he/she has in the real world.

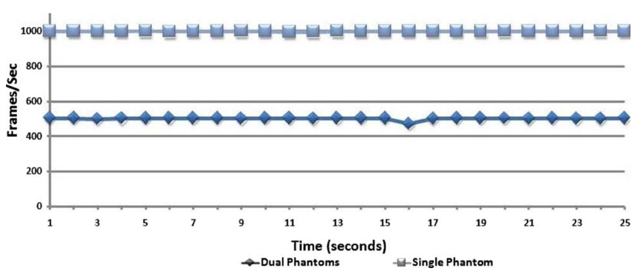

Fig. 5 Physics update rate during low clearance assembly

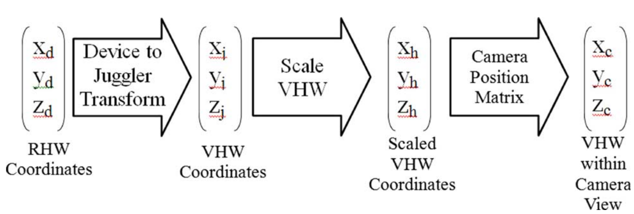

Fig. 6 Mapping RHW within camera view

7 Optimal Voxel Size Test

The system performance depends on the voxel size chosen for each mating part. Low clearance mating parts require a smaller voxel size for improved collision accuracy at the expense of memory and computation requirements. Figure 7 shows that the voxel size is exponentially proportional to the number of voxels (hence required memory). Figure 8 shows two CAD parts, a pin and a block, having a hole with a nominal diameter of 18.75 mm. We test the system for assembling the two parts with three different clearances: 2.5 mm, 1.4 mm, and 1.0 mm. For each clearance case, we first fixed the peg voxel size and varied the pin voxel size from 0.20 mm to 2.5 mm. The lower limit was chosen to be 0.20 mm due to the limitation of available computer memory. The operator was not limited by trial time, and it typically took less than 3 min to finish the assembly task. The results obtained from the assembly for each trial are recorded and analyzed. If the pin completely goes through the hole, the result was recorded as "yes." If the pin went only halfway through the hole, the result recorded was "half." For the remaining case, the result recorded was "no." All the tests were performed by the same operator.

Table 2 shows the result of assembly trials with a peg voxel size of 1.5 mm and a mating clearance of 2.5 mm. The test results indicate that smaller voxel sizes are not always the best choice. Using smaller voxel sizes results in creating a more accurate physics model representations. However, this leads to a greater number of pointshell-voxel interaction results in a "sticky" part behavior, adversely affecting system robustness. For the cases shown in Table 2, the optimal voxel size of the pin was [0.75,1.75] mm. A voxel size larger than 1.75 mm blocked the clearance, and a voxel size smaller than 0.75 mm caused vibration among parts. In either case, the assembly task could not be accomplished.

Figures 9–11 show the optimal pin voxel sizes for clearances of 2.5 mm, 1.4 mm, and 1.0 mm, respectively. It can be seen here that for higher clearances, a larger voxel size and a wider range of voxel sizes can be chosen. For instance, if the peg voxel size is chosen to be 1 mm, the pin voxel size range can be [0.25,1.8] mm when the clearance is 2.5 mm. However, this range drops to

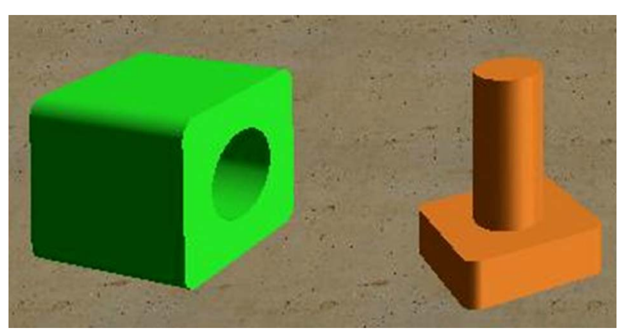

Fig. 8 Peg and hole

Table 2 Test assembly trials (clearance=2.5 mm and peg voxel size=1.5 mm)

Pin voxel size	No. of voxels of the pin	Result
0.25	113,850	No
0.5	28,416	Half
0.75	12,636	Yes
1	7024	Yes
1.25	4601	Yes
1.5	3190	Yes
1.75	2183	Yes
2	1820	Half
2.25	1360	Half
2.5	1172	No

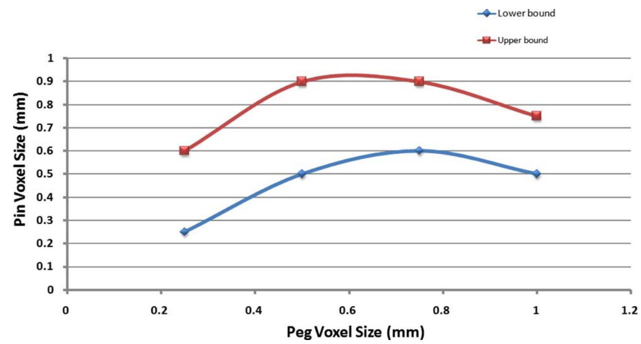

Fig. 11 Feasible pin voxel size (clearance=1.0 mm)

[0.5,0.75] mm for a clearance of 1 mm. In addition, the test showed that it is not possible to assemble the parts with a clearance of 0.5 mm no matter what voxel size was used.

8 Sharp Modules

The core capabilities in the SHARP system provide a platform that enables users to intuitively interact with complex CAD models and visualize rigid body dynamic behavior in an immersive environment using collision and physics behavior capabilities. Additional modules are designed and integrated into the SHARP system that takes advantage of these capabilities to allow designers to use VR for maintenance, training, and collaboration.

8.1 Swept Volumes. Modeling swept volumes is an effective way of resolving issues that may arise while servicing or inspecting complex mechanical assemblies. Questions related to accessibility, room for tooling, etc., for frequently serviced/replaced parts can be effectively answered using swept volumes during early stages of design. Within the SHARP environment, users can import components in an already assembled configuration and perform disassembly procedures to assess if there is enough room for accessibility, tooling, and parts. During the concept phase, engineers can create a swept volume (based on the path that the serviced component follows) and design other assembly components around it, ensuring space availability for maintenance tasks. SHARP uses the VPS voxel data to generate a swept volume by performing Boolean union operation on the voxel model being transformed during each motion frame (Fig. 12). The resultant VPS data are converted into a standard triangle format using a custom tessellation function. The data are then optimized using mesh optimization to create triangle data, which are visualized by the graphics scene-graph.

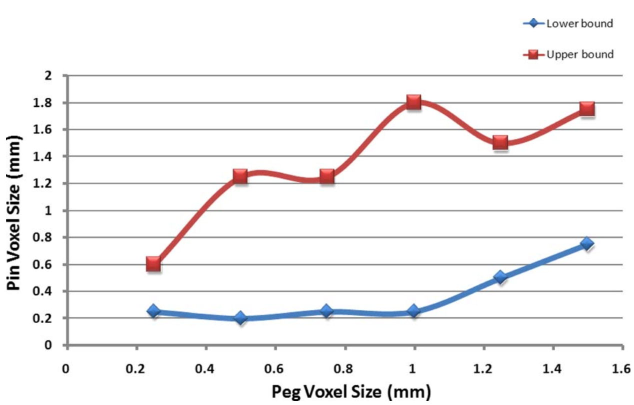

Fig. 9 Feasible pin voxel size (clearance=2.50 mm)

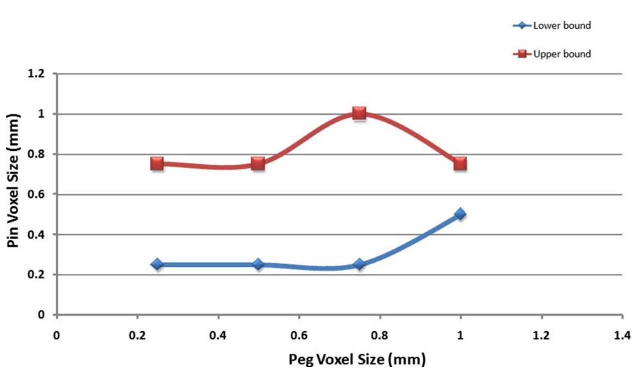

Fig. 10 Feasible pin voxel size (clearance=1.40 mm)

8.2 Record and Play Module. VR provides an ideal platform for tasks such as training assembly workers. Training workers in virtual environments can result in saving expensive down time on assembly lines. Immersive offline training can provide a more cost effective, interactive, and efficient way than conventional training techniques, which rely on paper manuals, video-based training, etc. Immersive training provides the user with a first-hand and more involving training experience, which holds promise for better procedure retention. The record and play capabilities allow users to record an assembly sequence performed by the operator. The sequence can then be displayed and analyzed several times

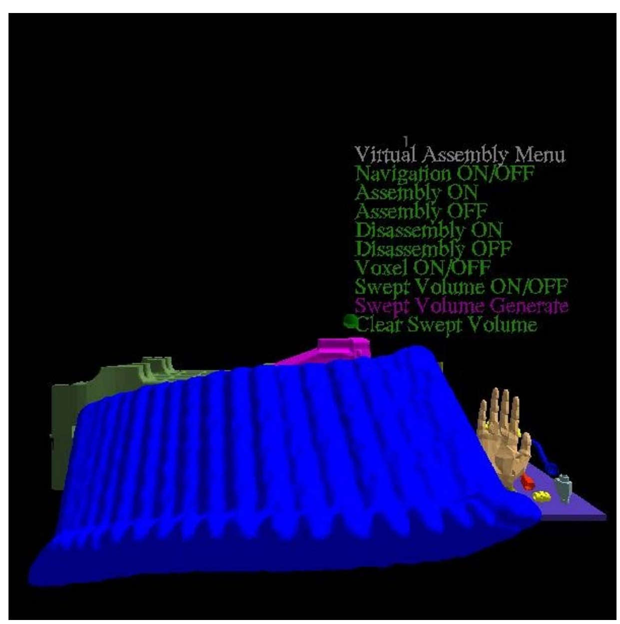

Fig. 12 Illustration of swept volumes in SHARP

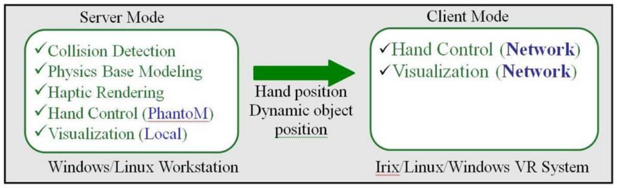

Fig. 13 Network architecture

within the virtual environment. During the recording phase, the system keeps track of all model transformations and stores them in a data file. During playback, the system reads the data file, repositions parts to their initial state, and displays the assembly sequences and part trajectories. This will facilitate assembly workers to better understand the designed assembly sequences and part trajectories before performing assembly operations in VR as well as the assembly line.

8.3 Subassembly Module. Subassemblies are an integral part of a mechanical assembly process. A mechanical assembly task can be any of the following:

- assembling two separate parts
- assembling a part with another subassembly
- assembling two subassemblies

SHARP supports the creation of subassemblies, which can allow training simulations of more comprehensive manual assembly processes. Performing dynamic assembly/disassembly operations in virtual environments requires modification of the underlying scene-graph or object hierarchy tree to maintain consistent object motions. When two or more parts are assembled together, their VPS data and display nodes are rearranged so that they behave as a single entity in the digital world. More details about the module design can be found in Ref. [9].

8.4 Network Module. The network module can be selectively activated in SHARP. When running in the network configuration, the application (running at the workstation with haptic feedback) acts as a server and communicates with the client application running at a geographically dispersed location. Figure 13 shows operations performed at the server and the client. The server runs in full mode; i.e., it loads graphic and haptic models and performs collision detection and physics-based modeling, calculates the model's final position, and sends the hand and dynamic model's position information to the client using TCP/IP. The client module loads graphic model representations and updates their transform based on the data received from the server.

9 Conclusions and Future Work

In this paper, a platform independent application, SHARP, has been presented, which uses physics-based modeling for simulating realistic part behavior and provides an intuitive dual-handed PHANToM® haptic interface for mechanical assembly in an immersive VR environment. SHARP is capable of assembling complex CAD geometry and supports a vast variety of VR systems for increased portability. Multiple modules are integrated into the system to perform service, maintenance evaluations, and virtual training.

The SHARP system demonstrates an attempt to successfully assemble complex CAD models by relying solely on the simulated physical constraints and haptic feedback. Users can import and assemble complex CAD components in a more realistic way without requiring part position or other proprietary CAD data. However, because the system uses voxel-based approximations for assembly, parts with low clearances cannot be assembled. In the future, methods for collision detection and physics modeling using accurate B-Rep surface representations will be examined for more memory efficient and highly accurate collision detection and physics computations. Also, combinations of constraint-based and physics-based methods will be explored to develop an optimum interaction paradigm, which can provide solutions to low clearance assembly, realistic part behavior, and haptic interactions.

Acknowledgment

We are grateful for the technical assistance of William McNeely of the Boeing Co. This work is funded by Deere and Co. This work was performed at the Virtual Reality Applications Center at Iowa State University.

References

[1] Banerjee, P., 2002, "Data Interface Software for Windows PC-Compatible Virtual Reality Scene Graphs," ASME J. Comput. Inf. Sci. Eng., **2**(1), pp. 72–74.
[2] Erleben, K., Sporring, J., Henriksen, K., and Dohlmann, H., 2005, *Physics-Based Animation*, 1st ed., Charles River Media, Hingham, MA, p. 817.
[3] McNeely, W. A., Puterbaugh, K. D., and Troy, J. J., 1999, "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," *Proceedings of the SIGGRAPH 99 Conference, Annual Conference Series*, Los Angeles, CA.
[4] Mirtich, B. V., 1996, "Impulse-Based Dynamic Simulation of Rigid Body Systems," *Computer Science*, University of California, Berkeley, p. 246.
[5] Baraff, D., 1995, "Interactive Simulation of Solid Rigid Bodies," IEEE Comput. Graphics Appl., **15**(3), pp. 63–75.
[6] Witkin, A., Gleicher, M., and Welch, W., 1990, "Interactive Dynamics," Comput. Graph., **24**(2), pp. 11–22.
[7] Fröhlich, B., Tramberend, H., Beers, A., Agarawala, M., and Baraff, D., 2000, "Physically-Based Modeling on the Responsive Workbench," *Proceedings of the IEEE Virtual Reality Conference*.
[8] McDermott, S. D., and Bras, B., 1999,"Development of a Haptically Enabled Dis/Re-Assembly Simulation Environment," ASME Paper No. DETC1999/CIE-9035.
[9] Seth, A., Su, H. J., and Vance, J. M., 2006, "SHARP: A System for Haptic Assembly and Realistic Prototyping," ASME Paper No. DETC2006/CIE-99476.
[10] Kim, C. E., and Vance, J. M., 2004, "Collision Detection and Part Interaction Modeling to Facilitate Immersive Virtual Assembly Methods," ASME J. Comput. Inf. Sci. Eng., **4**(2), pp. 83–90.
[11] Kuehne, R., and Oliver, J., 1995, "A Virtual Environment for Interactive Assembly Planning and Evaluation," *Proceedings of the ASME Design Automation Conference*, Boston, MA.
[12] Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., and Hart, P., 1999, "VADE: A Virtual Assembly Design Environment," IEEE Comput. Graphics Appl., **19**(6), pp. 44–50.
[13] Kim, C. E., and Vance, J. M., 2004, "Development of a Networked Haptic Environment in VR to Facilitate Collaborative Design Using Voxmap Pointshell (VPS) Software," ASME Paper No. DETC2004/CIE-57648.
[14] Marcelino, L., Murray, N., and Fernando, T., 2003, "A Constraint Manager to Support Virtual Maintainability," Comput. Graphics, **27**(1), pp. 19–26.
[15] Coutee, A. S., McDermott, S. D., and Bras, B., 2001, "A Haptic Assembly and Disassembly Simulation Environment and Associated Computational Load Optimization Techniques," ASME J. Comput. Inf. Sci. Eng., **1**(2), pp. 113–122.
[16] Wan, H., Gao, S., Peng, Q., Dai, G., and Zhang, F., 2004, "MIVAS: A Multi-Modal Immersive Virtual Assembly System," ASME Paper No. DETC 2004/CIE-57660.
[17] Gupta, R., Whitney, D., and Zeltzer, D., 1997, "Prototyping and Design for Assembly Analysis Using Multimodal Virtual Environments," Comput.-Aided Des., **29**(8), pp. 585–597, special issue on VR in CAD.
[18] Wang, Y., Jayaram, S., Jayaram, U., and Lyons, K., 2001, "Physically Based Modeling in Virtual Assembly," ASME Paper No. DETC2001/CIE-21259.
[19] Bullinger, H. J., Richer, M., and Seidel, K. A., 2000, "Virtual Assembly Planning," Human Factors and Ergonomics in Manufacturing, **10**(3), pp. 331–341.
[20] Fernando, T., Marcelino, L., Wimalaratne, P., and Tan, K., 2000, "Interactive Assembly Modeling Within a CAVE Environment," *Proceedings of the Eurographics–Portuguese Chapter*, pp. 43–49.
[21] Brough, J. E., Schwartz, M., Gupta, S. K., Anand, D. K., Kavetsky, R., and Petterson, R., 2007, "Towards the Development of a Virtual Environment-Based Training System for Mechanical Assembly Operations," Virtual Reality, **11**(4), pp. 189–206.
[22] Garbaya, S., and Zaldivar-Colado, U., 2007, "The Effect of Contact Force Sensations on User Performance in Virtual Assembly Tasks," Virtual Reality, **11**(4), pp. 287–299.

[23] Seth, A., Su, H. J., and Vance, J. M., 2005, "A Desktop Networked Haptic VR Interface for Mechanical Assembly," ASME Paper No. IMECE2005-81873.
[24] Just, C., Bierbaum, A., Baker, A., and Cruz-Neira, C., 1998, "VR Juggler: A Framework for Virtual Reality Development," *Proceedings of the Second Immersive Projection Technology Workshop (IPT98)*, Ames, IA. CD-ROM.
[25] Fifth Dimension Technologies, http://5dt.com/
[26] OPENGL PERFORMER, SGI, http://www.sgi.com/products/software/performer/
[27] Burdea, G. C., 1999, "Haptic Feedback for Virtual Reality," *Proceedings of the Virtual Reality and Prototyping Workshop*, Laval, France.
[28] Howard, B. M., and Vance, J. M., 2007, "Desktop Haptic Virtual Assembly Using Physically Based Modelling," Virtual Reality, **11**(4), pp. 207–215.
[29] Colgate, J. E., Stanley, M. C., and Brown, J. M., 1995, "Issues in the Haptic Display of Tool Use," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, PA.